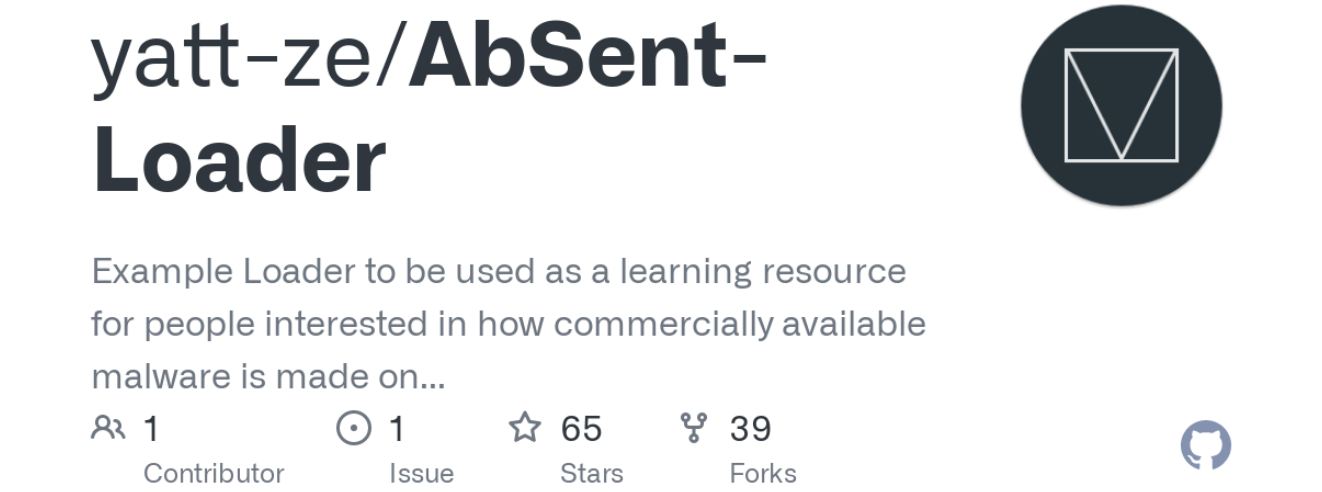
AbSent-Loader

github.com/Tlgyt/AbSent-Loader

yatt-ze



Example Loader to be used as a learning resource for people interested in how commercially available malware is made.

Join the discussion on discord: https://discord.gg/AMs6DA9

Definition of a loader

A "Loader" or "Dropper" is a type of malware not dissimilar to a botnet, usually built on the same C&C architecture they lack some of the more advanced features a fully featured botnet might have and instead try to be as lightweight as possible to be used as the 1st stage in an attack.

Many commercially available loaders extend their lifetime on the black market by going modular, providing updates and plugins that extend the loaders capability and provide the seller a larger revenue stream by selling the plugins separately from the main "Base" bot, these usually include but not limited too:

- DDOS Functions
- Password Stealing
- HRDP
- Web Injects
- Keyloggers

C&C Architecture

Many loaders and botnets, id say 90% nowadays use a PHP web panel for controlling the network, reasons being its easy to setup, provides a modest amount of security if done properly, and it looks pretty, allowing for graphs and maps of bots, nice pretty tables of executing tasks and client info, all makes a PHP panel for the C&C architecture a nice option, especially good for marketing (People like pretty things).

Unfortunately, or fortunity depending on the color of your hat, these panels are usually rather insecure, vulnerable to SQL injection and XSS, allowing for easy takeovers and shutdowns. So easy I've knowen people to exclusively build their botnet from others vulnerable panels, stealing all their bots and running a "Botkiller", basically an antivirus built into the client designed to detect and kill any competing malware on the infected system.

The architecture of these Php based control panels is very simple, they have a PHP file usually called something like "gate.php" or something not so obvious like "store.php", this page is the contact point for the client. The client will send a `POST` request (Some use `GET`) to the page containing the clients' information, and the page will respond with a command to execute. The way the commands are sent and phrased are different for every variant but is usually done with `JSON` or plain text. If done properly the page will verify the client is legit and make sure the supplied data isn't an XSS or an SQLi attack, and add it to the panel's database.

The Standard Client Loop

The client is what runs on an infected system, its job is simple, stay hidden and execute tasks.

On executing the client will try to "*Make itself at home*" that is, become persistent in the system, setting up defences to stop itself being killed and making sure its run when the system turns on again, it will also attempt to collect as much information about the computer it can, what version of the Operating System its running on, What privileges it has, the username, etc. It then gathers all this Information and sends it off to the C&C, receiving any tasks back and acting upon them. Some clients will try to be clever about the way it goes about this, commonly waiting for a while before actually executing anything to seem less suspicious.

Afterwards we enter the "*loop*" the client will go dormant for a set amount of time, usually around the 5 minute mark before reaching out for any new commands and letting the C&C know its still alive. Reason being to lighten the network load of the server and the infected system, the bigger the network, usually the longer the wait.

Disclaimer: I do not accept responsibility for the misuse of provided code blah blah blah don't be a cunt